

Demonstration of teleportation across a quantum network codeHjalmar Rall* and Mark Tame[✉]*Department of Physics, Stellenbosch University, Matieland 7602, South Africa* (Received 21 March 2024; accepted 2 July 2024; published 2 August 2024)

In quantum networks an important goal is to reduce resource requirements for the transport and communication of quantum information. Quantum network coding presents a way to do this by distributing entangled states over a network that would ordinarily exhibit contention. In this work, we study measurement-based quantum network coding (MQNC), which is a protocol particularly suitable for noisy intermediate-scale quantum devices. In particular, we develop techniques to adapt MQNC to state-of-the-art superconducting processors and subsequently demonstrate successful teleportation of quantum information, giving new insight into MQNC in this context after a previous study was not able to produce a useful degree of entanglement. The teleportation in our demonstration is shown to occur with fidelity higher than could be achieved via classical means, made possible by considering qubits from a polar cap of the Bloch sphere. We also present a generalization of MQNC with a simple mapping onto the heavy-hex processor layout and a direct mapping onto a proposed logical error-corrected layout. Our work provides some useful techniques to test and successfully carry out quantum network coding.

DOI: [10.1103/PhysRevA.110.022602](https://doi.org/10.1103/PhysRevA.110.022602)**I. INTRODUCTION**

Quantum communication enables interactions between physically distant quantum systems and opens the path to applications such as distributed quantum computing, quantum key distribution (QKD), and communication within quantum processors [1–3]. Significant progress towards practical quantum communication has been made in recent years: high-fidelity quantum communication between a single source and destination has been realized in a number of experiments, including free-space quantum communication [4], a large-scale QKD network with satellite link [5], and a three-node quantum network utilizing solid-state qubits [6], and research into optical communication between superconducting quantum computers is ongoing [7,8]. These systems are too small at present to be of practical use, but larger and more complex networks are becoming feasible, necessitating the study of quantum networks in a practical context. Of equally great importance are internal communication networks inside quantum processors, as the superconducting processors of Google, IBM, and Rigetti are rapidly growing to sizes where the standard entanglement and qubit swapping approaches become impractical [9]. In addition, methods for external communication networks that enable the linking up of small quantum processors to make an effective larger processor have started to gain attention recently [10,11], which is relevant in light of computational techniques for distributed processors such as entanglement forging [12].

Quantum networks have been studied at length in the literature [13–15], but many practical issues remain, especially in the current noisy intermediate-scale quantum (NISQ) era [16], where entanglement is imperfect and many rounds of purification may be required to achieve a sufficient degree

of entanglement. Thus the preshared entanglement required for a teleportation-based network is difficult to establish and the bandwidth of the network may be severely limited. Quantum networking is further restricted by the need for quantum routing which takes up valuable resources in terms of the number of qubits required and also introduces additional noise into the system. It is therefore necessary to find efficient schemes for quantum networking with limited qubit number and bandwidth. A solution to this is provided by quantum network coding (QNC) [17,18]. In classical networks with limited bandwidth, network coding [19–21] solves the problem of contention by encoding messages which must pass through bottlenecks and using uncontended channels to send decoding instructions. In certain communication scenarios, classical network coding can utilize all available bandwidth for useful communication despite the presence of bottlenecks. QNC mimics the classical case in that it makes use of local operations to achieve simultaneous transmission of messages through a bottleneck in a quantum network. In contrast to the classical case, this is achieved by redistributing the available channels (entanglement) so as to eliminate the bottleneck entirely.

QNC has experienced considerable interest since its introduction in 2007 [22], and has been studied both as a theoretical tool and as a practical protocol in quantum networks and processors [18,23]. It has also recently been demonstrated experimentally in an optical setup [24]. Measurement-based quantum network coding (MQNC) [25] is a very recent development which is well suited to the NISQ regime by virtue of requiring shallower circuits than existing QNC protocols. As a result of shorter circuit depth, the effect of qubit loss, gate errors, and qubit decoherence is reduced. MQNC has previously been studied on an IBM Q superconducting processor by Pathumsoot *et al.* [26], but the study was severely limited by the high degree of noise in the processor and did not realize quantum communication over the network code.

*Contact author: hjalmar.rall1@gmail.com

In the time since, IBM has made new processors available, with greatly reduced noise and a standardized layout which it is said will remain fixed for the foreseeable future. It is of interest then to see how MQNC performs on this new hardware and what further insight can be gained into practical implementation of MQNC beyond the general predictions made in the previous work. We overcome the challenges of translating MQNC to the new processor layout and show that—even with the extra overhead incurred—genuine quantum information transfer using teleportation over an MQNC network is possible on these processors, provided that the input states are restricted to a polar cap of the Bloch sphere, as in a recent theory proposal by Roy *et al.* [27].

With a view to the future, we also present a generalization of butterfly MQNC to a nonblocking network switch with an arbitrary number of nodes. Interestingly, the switch may be created directly on square grid topologies which are already in use or planned for use in superconducting quantum processors—examples include the Google Sycamore processor [28] and the planned error-corrected logical topology of the IBM processors [29]. While this is important for transferring quantum information within processors, it also has implications for networking within a quantum internet, where switching within quantum routers [30] is essential if entanglement between arbitrary pairs of end nodes is to be established using a shared physical link layer instead of private direct connections between nodes.

The paper is structured as follows: In Sec. II, we give a brief overview of measurement-based quantum computing, QNC, and MQNC, and introduce the generalized switch. In Sec. III, we present the particulars of the protocol and the method used to adapt the previous work to a newer processor. Section IV forms the main body, where the results of teleportation using MQNC are presented. Section V presents a general mapping of MQNC onto IBM processors. We end with a discussion and concluding remarks in Sec. VI.

II. BACKGROUND

A. Measurement-based quantum computing

A graph state is an entangled state $|G\rangle$ with qubits and entanglement between qubits corresponding to the vertices and edges of an undirected graph $G = (V, E)$. An N -qubit graph state with edge set E is defined according to

$$|G\rangle = \prod_{\{i,j\} \in E} CZ_{i,j}|+\rangle^{\otimes N}, \quad (1)$$

where CZ_{ij} is the controlled phase operation on qubits i and j , and $|+\rangle$ is the Pauli- X eigenstate with eigenvalue $+1$. A number of quantum operations transform between graph states and can thus be viewed as operations on the underlying graph. The following are the most common transformation rules that will be used in this work [31]:

(i) T1: A Z -basis measurement on a qubit a removes the corresponding vertex and incident edges from the graph.

(ii) T2: A Y -basis measurement on a qubit a removes the corresponding vertex and incident edges, and complements the subgraph induced by the neighborhood N_a , i.e., $G(V, E) \rightarrow G(V/\{a\}, E \Delta K)$, with K the edge set for the complete subgraph induced by $N_a \cup \{a\}$ and Δ the symmetric

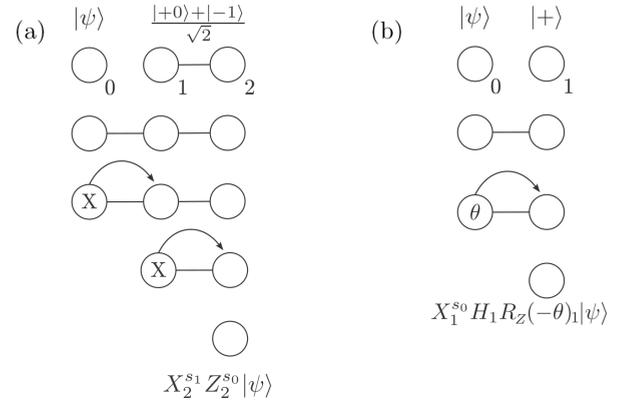


FIG. 1. (a) A state $|\psi\rangle$ may be teleported across a linear section of a graph state. (b) Teleportation can be used to perform the operation $HR_Z(-\theta)$. X 's within qubits indicate X -measurements; θ indicates a measurement in the basis $\{\frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta}|1\rangle), \frac{1}{\sqrt{2}}(|0\rangle - e^{i\theta}|1\rangle)\}$.

difference. In other words, neighbors of a are connected unless a connection already exists, in which case it is broken.

(iii) T3: X -basis measurements on two adjacent qubits a and b removes them and complements the bipartite subgraph induced by N_a and N_b , i.e., $G(V, E) \rightarrow G(V/\{a, b\}, E \Delta K)$, with K the complete bipartite subgraph induced by $N_a \cup \{a\}$ and $N_b \cup \{b\}$. In other words, all the neighbors of a are connected to all the neighbors of b unless a connection already exists, in which case it is broken.

(iv) T4: Local complementation on a qubit a given by $\sqrt{X_a} \prod_{b \in N_a} \sqrt{Z_b} |G\rangle$ complements the subgraph induced by N_a and leaves a and its incident edges unchanged. In other words, neighbors of a are connected unless a connection already exists, in which case it is broken. This is a nondestructive version of the Y -measurement transformation rule T2 where the qubit a is not removed from the graph. It should be noted that Pauli byproducts are introduced for certain measurement outcomes for the above operations. These must be tracked by a classical computer and either adaptively corrected or commuted through to the end of the circuit before being corrected (possibly via postselection). For an in-depth discussion of graph states, we refer the reader to Ref. [31]. Graph states form a resource for measurement-based quantum computation (MBQC) [32,33], many elements of which are used in MQNC. An arbitrary single-qubit state $|\psi\rangle$ may be attached to a graph state using a controlled phase gate. This state may subsequently be transported within the graph by means of quantum teleportation along linear sections. An example of this is shown in Fig. 1(a): First, a two-qubit (linear) graph state $\frac{1}{\sqrt{2}}(|+0\rangle + |-1\rangle)_{12}$ is created, then the state $|\psi\rangle_0$ to be teleported is entangled with this state via a controlled phase gate, and, finally, the measurements $M_{x,0}$ and $M_{x,1}$ in the basis $\{\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\}$ with outcomes s_0 and s_1 , respectively, are performed. This yields the state $X_2^{s_1} Z_2^{s_0} |\psi\rangle_2$ which may be transformed back to $|\psi\rangle$ if the measurement outcomes are known.

Arbitrary unitary operations may be performed on a single-qubit state attached to a graph state by way of appropriate projective measurements, which serve to both teleport and transform the single-qubit state. An example is shown in

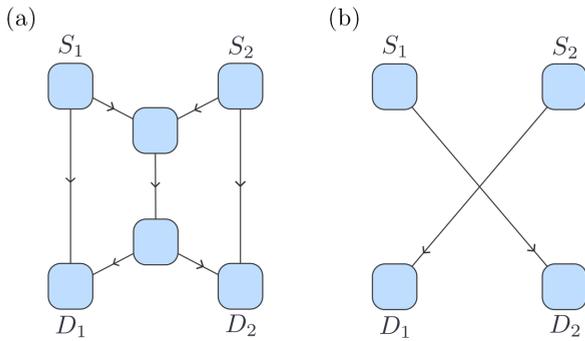


FIG. 2. (a) The butterfly network and (b) the new network after performing QNC. Blue shading represents network nodes. Lines represent quantum communication channels. Classical communication is assumed to be free.

Fig. 1(b): First, the state $|\psi\rangle_0$ is entangled with the graph state (here, the single-qubit graph $|+\rangle_1$) via a controlled phase gate, then a measurement in the basis $\{\frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta}|1\rangle), \frac{1}{\sqrt{2}}(|0\rangle - e^{i\theta}|1\rangle)\}$ (with θ an arbitrary angle) is performed on qubit 0. This yields the state $X_1^{s_0} H_1 R_Z(-\theta)_1 |\psi\rangle_1$. Such operations may be composed to obtain arbitrary unitary operations. Given a sufficiently large two-dimensional (2D) grid (cluster) graph state, measurement-based computation is universal. Since all two-qubit operations are performed during the creation of the resource state, and they are all commutative, they may be done simultaneously if the hardware allows. Since all byproducts are Pauli operators, they can be commuted through to the end of the circuit either directly or adaptively and subsequently simplified, leading to further decreases in circuit depth.

B. Quantum network coding

Quantum network coding is best illustrated through the example of the butterfly network. Given the network shown in Fig. 2(a) with each channel having capacity 1, the goal is to simultaneously send qubits from S_1 to D_2 and from S_2 to D_1 . In Ref. [22], it is shown that it is not possible to do this perfectly if only quantum communication is allowed. It was later shown that perfect quantum network coding is possible if free classical communication is allowed [18], as shown in Fig. 2(b). Protocols for perfect QNC have been developed for the case where transmitters share entanglement [17] and for the case of the butterfly network across quantum repeaters [23]. The former protocol has been demonstrated experimentally in an optical setup [24] with sufficient fidelity to enable teleportation of quantum information with fidelity exceeding the classically achievable bound. These protocols, however, require complex circuits and additional steps for resource state creation. On the other hand, MQNC [25] presents a measurement-based alternative to the repeater network protocol with a reduction in circuit depth of 50% and a corresponding increase in the allowable gate error to achieve a specified fidelity. Furthermore, this protocol contains, as an intermediate step, a graph state which also has applications in on-processor teleportation.

The protocol proceeds as follows: Starting with seven Bell pairs, a six-qubit graph state is generated as shown in Fig. 3(a)

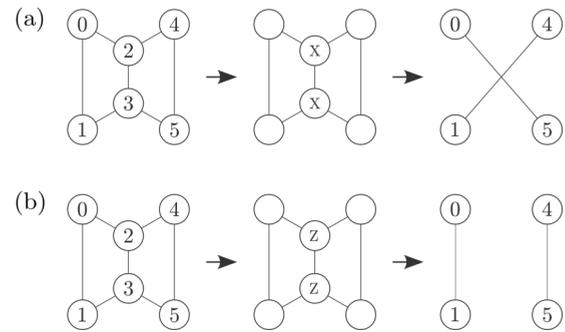


FIG. 3. The procedure for MQNC starting with a six-qubit graph state. (a) and (b) show the two different configurations (cross pairs and straight pairs) of MQNC. Circles and lines (which are vertices and edges of graphs, respectively) represent qubits and entanglement between qubits.

on the left-hand side. This state may also be generated directly via controlled phase gates according to Eq. (1). By measuring the two central qubits in the X-basis [Fig. 3(a), middle], the entanglement in the graph state can be redistributed so as to give two cross pairs (up to Pauli byproducts) using transformation rule T3 or, alternatively, by measuring the central qubits in the Z-basis; the state shown on the right-hand side of Fig. 3(b) is created (up to Pauli byproducts) using transformation rule T1. The two-qubit graph states generated by the MQNC measurements are maximally entangled and can be used for teleportation, where qubits are first entangled with the state generated by MQNC and subsequently teleported to the destination nodes, as shown in Fig. 4. Here, the states to be teleported may also be entangled before the MQNC protocol begins, as the entangling commutes with the measurements of the central qubits. Effectively, the three vertical channels with a bottleneck along one of the possible desired routes have been used to create two possible configurations through which the desired quantum communication routes may be directly achieved.

C. Generalized MQNC

MQNC has thus far been considered in the same context as traditional classical network coding, where the primary goal is to increase network throughput. Quantum networks, however, are limited to the multiple unicast scenario (k -pair problem [34,35]) due to no-cloning [36], and therefore the primary benefit of QNC is that it solves contention and therefore succeeds in limited capacity networks where quantum routing fails. The problem of communicating within a quantum network ultimately reduces to simultaneously connecting k pairs of source and destination qubits. This is more closely related to switching than to routing, and indeed butterfly QNC acts like a 2×2 nonblocking switch.

Motivated by this interpretation, before proceeding with the details of our demonstration, we present a generalization of MQNC in Fig. 5. Unlike the previous generalization in Ref. [35], our scheme requires no more than four connections per qubit, regardless of the number of source and destination nodes. We have not seen such a construction in the literature [18,22,25]. A similar quantum switching network was studied

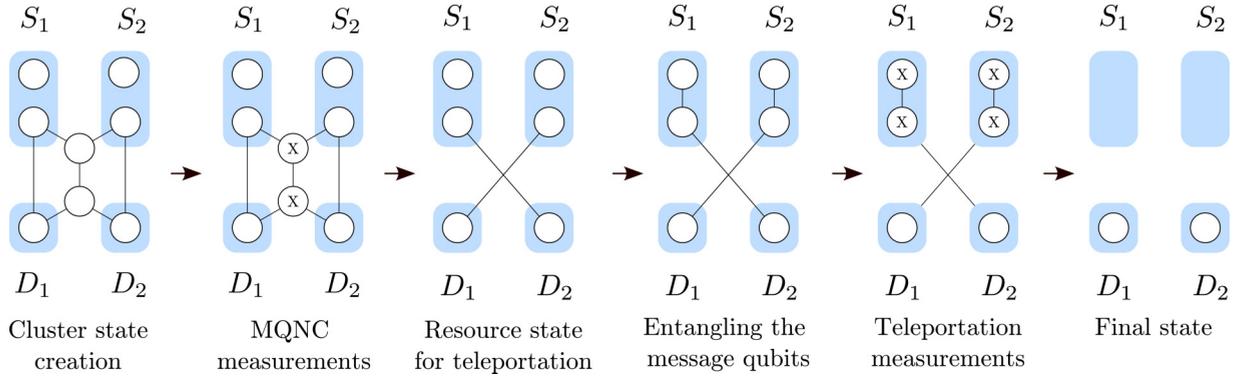


FIG. 4. The procedure for performing MQNC to generate cross pairs and subsequently teleporting states across these pairs. The straight pair (Z-measurement) case outlined in Fig. 3(b) is similar. White circles represent qubits and blue shaded regions indicate a single source or destination node in the network.

in Ref. [37], but the 2×2 switches in that network were based on CNOT and Toffoli gates and thus substantially differ from the measurement-based approach presented here. Our switching network consists of multiple copies of the MQNC code joined together, as shown in Fig. 5(a), in a Spanke-Beneš network [38]. This network is chosen because it is an optimal nonblocking planar network, with the planar property being essential for two-dimensional quantum processor topologies which cannot directly implement crossed connections as in a standard nonplanar Beneš network. The network coding measurements create linear graph states between source and destination, as shown in Fig. 5(b), and the intermediate qubits in these states are subsequently removed via Pauli-Y measurements using transformation rule T2. If the switch is centrally located and each source or destination qubit is at a separate location, only $2k$ nonlocal connections to the switch are required.

Our generalized MQNC scheme uses a total of $\frac{k(k-1)}{2}$ switches, which was shown in Ref. [38] to be the smallest

number for a nonblocking planar network. Since there is some overlap between switches, the total number of required qubits is only $4k(k-1) + k$. The number of 2×2 MQNC switches can be reduced while keeping the nonblocking property by making use of an arbitrary-size Beneš network [39], which uses only $O(k \log_2 k)$ 2×2 switches, but this might be difficult to implement on two-dimensional quantum processor topologies since Beneš networks [40] have crossed connections (i.e., they are nonplanar). The Spanke-Beneš network, on the other hand, has a structure which is well suited to these topologies and, in fact, it may be directly created given a square grid topology. We show later how it may also be easily created on the so-called heavy-hex topology currently in use on IBM quantum processors.

III. MQNC ON IBM PROCESSORS

Previously, Pathumsoot *et al.* [26] studied MQNC on the superconducting processor IBM Q 20 Tokyo by generating the

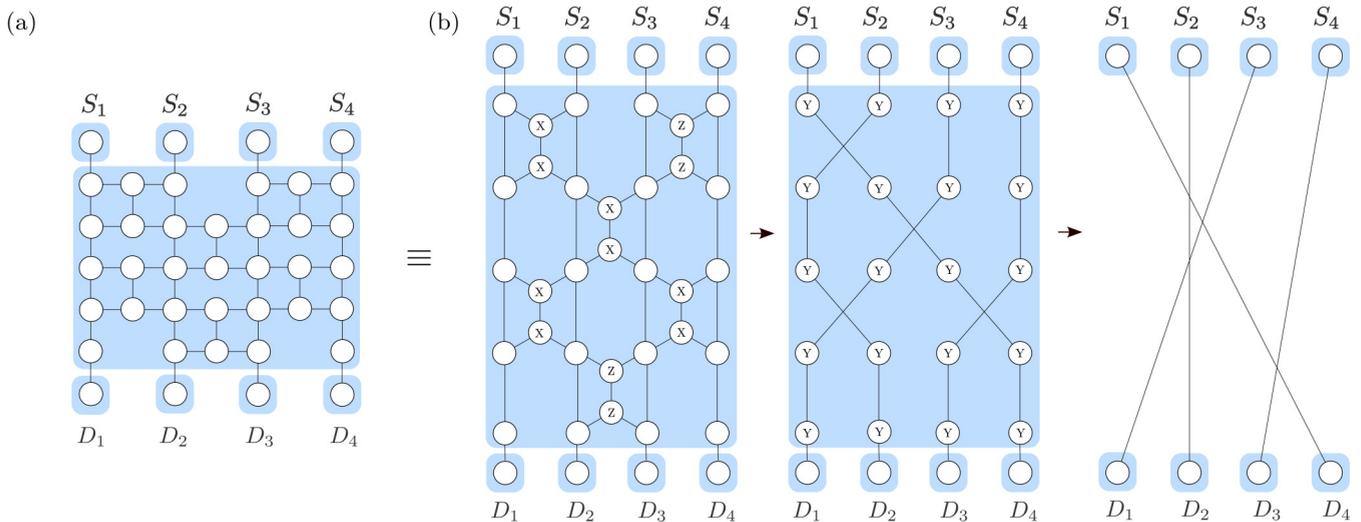


FIG. 5. A measurement-based quantum switch utilizing MQNC. A Spanke-Beneš network is chosen because it is optimal for planar networks. The square grid structure is showcased in the 4×4 switch shown in (a). In (b), the same network is expanded to show how it is made up of butterfly networks. The measurements in (b) give an example of how MQNC can be used to achieve a particular permutation of connections between four sources and four destinations. After performing the MQNC measurements, Y-measurements are used to remove residual qubits according to transformation rule T2.

six-qubit resource graph state shown in Fig. 3 (left-hand side) directly using the two-qubit gates available on the processor. They showed that entanglement existed in the resource state and reported on the fidelity and concurrence of the final two-qubit states, achieving fidelities of 0.57 ± 0.01 and 0.58 ± 0.01 for the cross pairs (fidelities for straight pairs were not reported). Lastly, they tested for violation of Clauser-Horne-Shimony-Holt (CHSH) inequality in the final states, but found none due to deterioration of the state from processor noise. It was determined that CHSH violation should be possible with single-qubit gate errors that are approximately half those of IBM Q Tokyo.

The IBM superconducting processors are devices undergoing rapid development, and the past two years have seen dramatic decreases in noise and errors. One such processor is *ibm_cairo*, a 27-qubit device of the “falcon” series. As shown in Table I in the Appendix, it has significantly reduced error compared to IBM Q 20 Tokyo, the error rates for which are reported in Table II in the Appendix. As with all current IBM processors, however, this reduction in error rates comes at the cost of a processor topology wherein the hardware for two-qubit operations is more sparsely distributed among the qubits. While this processor topology has definite advantages and is planned to be IBM Quantum’s standard for the foreseeable future, it precludes the possibility of creating the six-qubit resource state using direct entangling operations, as was done with a specialized processor topology in [26]. In light of this, we consider it prudent to adapt the demonstration of Pathumsoot *et al.* to these newer processors, to show that the connectivity limitations of the improved processors can be overcome and subsequently to show stronger evidence that MQNC will become practical on near-future NISQ devices.

We performed MQNC as outlined in Fig. 4 on the IBM Q falcon superconducting processor *ibm_cairo*. The processors were remotely accessed through the IBMQ API and the QISKIT PYTHON library. In place of standard circuit transpiling techniques utilizing excessively noisy SWAP operations, we employ graph-state rewiring techniques in order to overcome the processor topology limitations; a discussion of this follows in the next section. We subsequently demonstrate that the resource required for performing MQNC can be generated with fidelity exceeding that obtained in the prior work. Despite this, it is still not possible to perform full teleportation across the resource using MQNC. However, we show that by considering the teleportation of a region of states from the Bloch sphere, the utilization of quantum correlations during the teleportation can be confirmed, thus demonstrating the use of quantum effects in MQNC.

Processor topology

All current IBM Q processors make use of variations on the layout shown in Fig. 6. This layout was chosen in anticipation of future generations of the processors which will have sufficiently low gate error to implement error-correction codes (ECCs) native to this layout [29]. These are stabilizer codes with logical qubits arranged in a square lattice, where two-qubit operations may be performed between nearest neighbors. Consequently, they would also allow for the creation of error-corrected or “encoded” graph states based

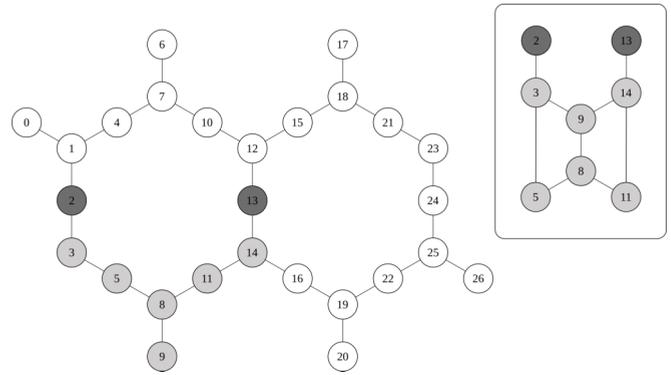


FIG. 6. Layout of *ibm_cairo*. Edges indicate neighboring qubits on which CNOT gates may be performed. Light-gray qubits are used to create the six-qubit graph state for MQNC. Dark-gray qubits are teleported across the network code graph state by entangling them with the graph state and subsequently performing appropriate measurements. Inset: The new layout after rewiring has been performed.

on the square grid. It is interesting to note that—since graph states are stabilizer states—such an encoded graph state may be viewed as a single stabilizer state of the physical qubits which encodes multiple logical qubits in its stabilized subspaces. This is related to the work in Refs. [41–43], where it was shown that there is a one-to-one correspondence between stabilizer codes and graph codes.

In current processors, where the gate error is still too high for implementation of the ECCs in Ref. [29], physical qubits must be used and we are limited by the reduced connectivity of the heavy-hex lattice (compared to the square lattice) which is not well suited to the creation of arbitrary graph states. Nonetheless, direct graph-state generation on the heavy-hex lattice is possible. Nonadjacent qubits may be entangled by the use of SWAP operations, and the QISKIT transpiler provides functionality to automatically determine the necessary operations and perform optimization. While the QISKIT transpiler works in general, it fails to produce a practical circuit for the generation of a six-qubit graph state, as the large number of two-qubit gates introduces an excessive amount of noise. During the initial stages of our study, we attempted to generate the six-qubit graph state using SWAP-based transpiling. However, we found that this resulted in states close to completely mixed states for the resulting pairs due to the large number of required two-qubit gates that introduce an excessive amount of noise. More advanced transpilers are available (see, for instance, those used in Refs. [44] and [45]), but these are also not tailored to graph-state generation.

For example, in order to entangle an arbitrary pair of qubits, it is first necessary to apply SWAP operations until the states occupy adjacent physical qubits. Doing so incurs a significant amount of noise, as each SWAP operation consists of three CONTROLLED-NOT gates with an average error of the order of 4×10^{-2} . Furthermore, the calculation of an optimal sequence of SWAP gates incurs a nontrivial classical processing cost. Attempting to transpile the MQNC circuit to the processor using the built-in tools of the QISKIT package results in a circuit incorporating a minimum of 12 noisy two-qubit gates, and results in final states close to the maximally mixed

state. Effectively, the new processors are no better than the old one at performing MQNC when this transpiling scheme is used. Here we present an alternative transpiling scheme based on local complementation (transformation rule T4) which requires only seven two-qubit gates and yields a circuit of reasonable depth. This means that the transpiling step, while still noisy, is no longer so noisy as to completely negate the improvements in hardware in the new processors.

Let a be a qubit in a graph state $|G\rangle$ with graph (G, E) . Let N_a be the neighborhood of a in $|G\rangle$. The local operation $LC_a = \sqrt{X_a} \prod_{b \in N_a} \sqrt{Z_b} |G\rangle$, known as local complementation, produces a new graph state with graph $G'(V, E')$ with $E' = E \Delta E(N_a, N_a)$ following transformation rule T4. As an example of using local complementation to redistribute entanglement, consider three qubits $a, b,$ and c in a graph, of which only a and $b,$ and b and $c,$ may be directly entangled via an edge. Entanglement between a and c can be created by performing local complementation on b . This leaves all three qubits entangled. Any one edge in this new triangle graph may be removed by performing local complementation on the opposite qubit. Any two edges may be removed by performing a Z -measurement on their common qubit, which has the effect of deleting all incident edges on that qubit following rule T1. In this way, an operation requiring six two-qubit operations using SWAP gates is achieved using only two with local complementation.

In order to generate the six-qubit graph state for MQNC, we implement the following sequence of operations (from right to left), which are shown in Fig. 7:

$$\begin{aligned}
 & LC_5 CZ_{1,3} CZ_{3,5} LC_3 LC_4 \\
 & \times LC_5 CZ_{3,5} CZ_{4,5} LC_2 LC_3 \\
 & \times LC_0 LC_1 CZ_{2,3} CZ_{1,3} CZ_{1,0},
 \end{aligned}$$

where the logical qubits 0, 1, 2, 3, 4, and 5 are mapped onto the qubits 5, 3, 8, 9, 11, and 14 on the processor, as shown in Fig. 6, as these represent the best least-error-prone set of qubits with the appropriate shape and having no qubit which is particularly error prone. Qubits 2 and 13 are then the states to be teleported across the resource.

IV. IMPLEMENTATION ON NEW PROCESSORS

We implemented MQNC on `ibm_cairo` using the rewiring scheme introduced above. A single run constitutes calibration for readout error mitigation, state tomography of the six-qubit resource state, state tomography of one of the two-qubit graph states generated using MQNC, and process tomography of teleportation across the same two-qubit graph state. This is repeated for each of the four pairs in turn, and the whole procedure is then repeated 30 times to account for variation in processor noise. The runs were spread out between 9:00 p.m. on April 26, 2022 and 5:00 a.m. on April 27, 2022 (UTC) due to use of the fair-share queuing system, and the daily calibration data are given in the Appendix. The variation in processor noise is found to be sufficiently small over a timescale of minutes, and the delay between readout error mitigation calibration and demonstration is not significant, so that results are not skewed from one pair to the next.

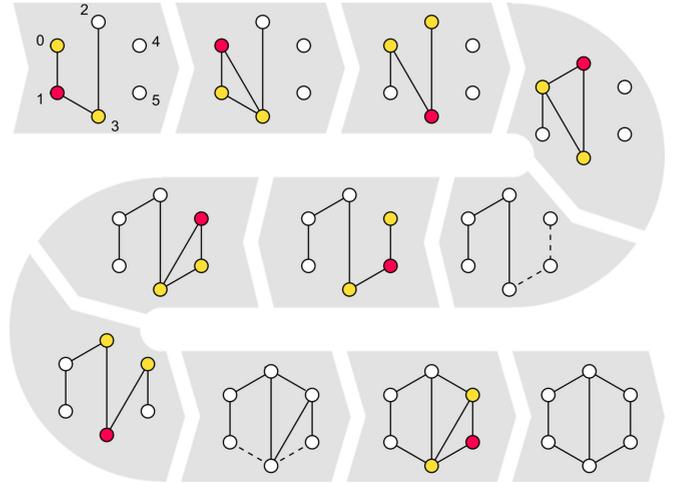


FIG. 7. The procedure for “rewiring” the processor layout to obtain the six-qubit resource state for the network code. We start by creating the three edges in the first graph. Local complementation is performed on pink qubits. Yellow qubits are in the neighborhood of the latter. Dashed lines indicate further entangling operations.

The measurements in the protocol result in probabilistic byproduct operations on the desired final state. At the time of use, the IBM processors did not support the feed-forward functionality necessary to undo these operations (this is expected in the near future), so for testing purposes, results must be postselected based on measurement outcomes. The four possible outcomes of two measurements at the central qubits of the network code correspond to four possible byproducts (including the identity) on each of the final two-qubit states, which occur with equal probability so that 1/4 of the results are kept. The measurements required for the teleportation also result in byproduct operations, but these are either one of the operations in $\{I, X, Z, XZ\}$ with equal probability so that combined with the byproducts from the network coding, we have the identity byproduct on the final teleported qubit with probability 1/4, and 1/4 of the results remain after postprocessing.

A. Genuine multipartite entanglement

The demonstration of genuine multipartite entanglement (GME) on IBM processors has proved challenging so far. In particular, a recent study attempting to create a 20-qubit linear graph state on `ibmq_poughkeepsie` [46] found that no subset of more than three qubits had undeniable GME. To confirm that the six-qubit graph state for MQNC can be successfully generated, we perform the measurements necessary to calculate a fidelity-based GME witness [47]. First note that graph states are stabilizer states with stabilizers for an N -qubit state given by $S_i = X_i \prod_{j=0}^{N-1} Z_j^{(\Gamma_{ij})}$, where Γ is the adjacency matrix for the corresponding graph. The graph state is uniquely defined by these stabilizers and explicitly given by $\rho = |G_{2 \times 3}\rangle\langle G_{2 \times 3}| = \prod_{i=0}^5 \frac{1+S_i}{2}$ [24]. Multiplying out this product, we get a sum over 64 projectors. Given the expectation value of each, the fidelity may be calculated and from that, it can be determined if GME exists. As in the article by Pathumsot *et al.* [26], we define fidelity

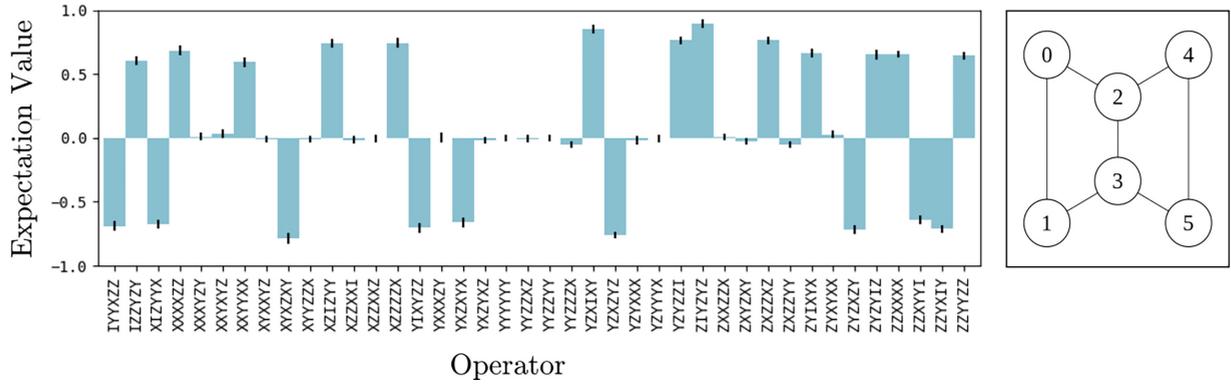


FIG. 8. Expectation values of the stabilizers of the six-qubit graph-state resource. The convention used is that the tensor product operators read from left to right, with the leftmost acting on qubit 0 and the rightmost on qubit 5.

according to

$$F(\rho, \sigma) = (\text{Tr}[\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}])^2 \\ = \text{Tr}(|G_{2 \times 3}\rangle\langle G_{2 \times 3}| \sigma),$$

where the final state and expected state have density matrices σ and ρ , respectively. Since the expectation value of the identity is simply 1, the expectation value of any projector of ρ acting as the identity on one qubit of σ may be determined without reference to that qubit. Hence multiple measurements may be combined into one and only 40 local measurement settings are required to obtain the full set of 64 expectation values. Here, 2000 shots are used per measurement basis. The results are shown in Fig. 8. A fidelity of 0.74 ± 0.02 was obtained.

Tóth and Gühne [47] give the witness $\langle \hat{\mathcal{V}} \rangle = \alpha \mathbb{1} - |G\rangle\langle G|$ for GME near a graph state, where α is defined to be the maximum overlap of the state $|G\rangle$ with any bipartition of qubits. If the witness is negative, then the overlap of the state with the ideal one is greater than the overlap of any bipartition with the ideal state, and hence the state is not biseparable. Therefore, $\langle \hat{\mathcal{V}} \rangle$ is a GME witness. The routine provided in QUBIT4MATLAB [48] was used to calculate α and it was found that $\alpha = 0.5$, leading to a negative $\langle \hat{\mathcal{V}} \rangle$ of -0.24 ± 0.02 with a large margin below zero. We conclude that the implementation of the six-qubit graph state has GME and, together with the high fidelity, this indicates a largely successful creation of the graph state.

B. Two-qubit state tomography

In order to measure the quality of the generated state and compare to the results of the previous study [26], we perform quantum state tomography [49] on each of the four pairs that can be generated using MQNC, as shown in Figs. 3(a) and 3(b) on the right-hand side. Readout error mitigation is then applied to the results and they are subsequently post-selected so as to correspond to the byproduct-free graph state. Here, 4000 shots are used for each of the nine tomography circuits, so that approximately 1000 shots remain after post-selection. Density matrices are obtained from the tomography results, and the fidelity compared to the ideal state $|G_2\rangle = \frac{1}{\sqrt{2}}(|0+\rangle + |1-\rangle)$, as well as the purity and concurrence, are calculated. Concurrence is an entanglement monotone defined

according to

$$C(\rho) = \max\{0, \lambda_1 - \lambda_2 - \lambda_3 - \lambda_4\},$$

where the λ_i are the square roots of the eigenvalues of $\tau = \rho(Y \otimes Y)\rho^*(Y \otimes Y)$ in descending order. The results are shown in Fig. 9. It is interesting to consider a potential connection between the quality of the pairs and the error rates of the corresponding qubits of the processor. The ‘‘protocol’’ qubits 0, 1, 4, and 5 (see Fig. 8) correspond to ‘‘processor’’ qubits 3, 5, 14, and 11 (see Fig. 6), and from best to worse for the fidelity of the protocol qubit pairs in Fig. 9 we have (1,4), (1,0), (0,5), and (4,5). From this, we may infer that protocol qubit 5 (processor qubit 11) may be the worst qubit, followed by protocol qubit 0 (processor qubit 3), then protocol qubit 4 (processor qubit 14), and, finally, protocol qubit 1 (processor qubit 5). In other words, in terms of a contributing factor to the low fidelity of the pairs, we have, for the processor qubits, the hierarchy $5 < 14 < 3 < 11$. However, by analyzing the error rates in Table I of the Appendix and noting the single-qubit

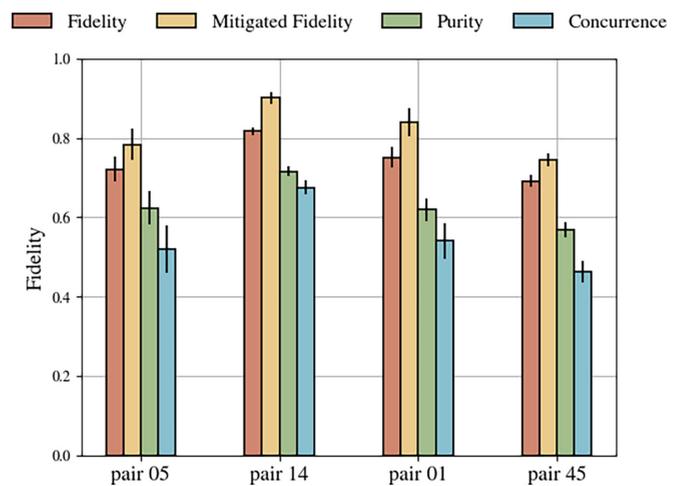


FIG. 9. Results of full state tomography on the two-qubit entangled states generated using the MQNC protocol on `ibm_cairo`. Fidelity is shown before and after application of readout error mitigation on the two qubits. Purity is defined as $P = \text{Tr}(\rho^2)$. The protocol qubits 0, 1, 4, and 5 (see Fig. 8) correspond to processor qubits 3, 5, 14, and 11 (see Fig. 6), respectively. Single- and two-qubit error rates are given in Table I in the Appendix for reference.

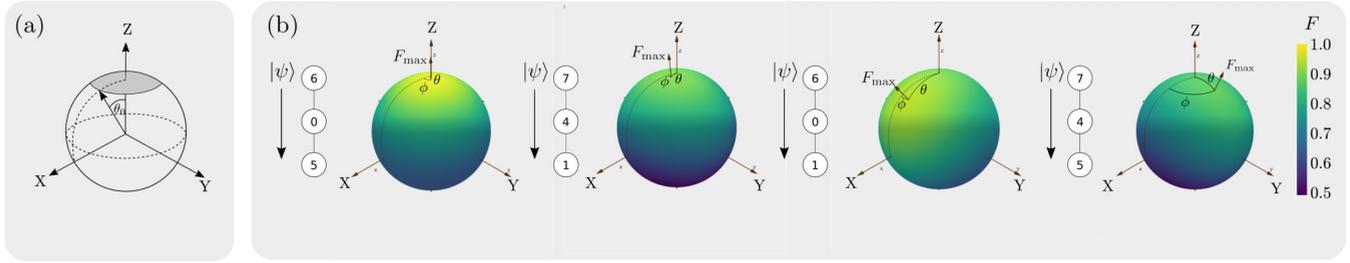


FIG. 10. A polar cap of the Bloch sphere is shown in (a) along with its angular radius θ_0 . (b) The teleportation fidelity as a function of initial state for each of the four teleportation routes. The highest fidelities fall within a spherical cap of the Bloch sphere (i.e., a polar cap rotated through the angles θ and ϕ). The arrow F_{\max} crosses the surface at the point of highest fidelity. The state to be teleported is initially encoded in the qubit labeled 6 (for the case of 05 and 01) or 7 (for the case of 41 and 45), where the protocol numbering from Fig. 3 is used, rather than the processor numbering.

error rates, we have the error-rate hierarchy $14 < 3 < 11 < 5$, which does not match up with the fidelity of the pairs.

On the other hand, the two-qubit CX gate error for processor qubit pair (11,14) is the worst and it may explain why this pair [protocol qubit pair (4,5)] is the worst. However, due to the local complementation sequence used to generate the graph state, shown in Fig. 7, the impact of this two-qubit gate is not easy to track. But, it could be a main contributing factor. Further detailed work on the propagation of noise would help in assessing how different single-qubit and two-qubit error rates impact the fidelity of the final pairs in the network code using the generation method that we have proposed.

Despite the noise incurred by the entire rewiring process, `ibm_cairo` shows a significant improvement in both fidelity and concurrence over the results of Pathumsoot *et al.* [26]. Notably, the cross pairs show an improvement in concurrence by a factor of 2 compared to 0.25 ± 0.02 and 0.36 ± 0.02 , respectively, obtained by Pathumsoot *et al.* This proves that a much higher degree of entanglement has been established in our final state.

C. Teleportation

We next tested whether the improved quality of `ibm_cairo` over earlier processors allows MQNC to achieve its intended purpose of generating a state over which quantum information can be transferred. For each of the pairs in the quantum network code in turn, the graph state is generated using the LC procedure introduced earlier, and the MQNC measurements that generate the pair are performed. The state to be teleported is then encoded in a qubit adjacent to the first qubit of the pair, entangled with the first qubit of the pair, and the Pauli- X measurements to teleport it onto the second qubit are performed, as shown in Fig. 4 for the case of cross pair generation.

Results are postselected based on MQNC and teleportation measurement outcomes. Quantum process tomography is performed on this circuit and a Choi matrix representation of the teleportation channel is obtained [50]. We then use the Choi matrix to obtain the fidelity of sending different states through the teleportation channel. The standard basis is used, with 2000 shots per circuit. We collect 30 Choi matrices for each channel to obtain a mean and error bar on the extracted fidelity.

In the current context, the teleportation can be viewed as a game, where Alice wishes to communicate a quantum state

to Bob without physically sending the state, and where Alice is allowed to perform a measurement on the state. A natural question which arises is whether it is possible to do so using only classical communication and no quantum correlations. While it is clearly impossible to gain enough information from a state with a single measurement to be able to perfectly recreate the state after classical transmission of the information, it is nonetheless possible to gain some information. This bound on the fidelity of classical “teleportation” is studied in Refs. [51] and [52]. For there to be any benefit to quantum teleportation and for confirming that quantum correlations have been used, the fidelity must exceed this classical bound [52,53].

The usual measure for teleportation fidelity is the average gate fidelity between the teleportation channel and the identity gate, defined according to the integral $F_{\text{ave}}(\epsilon, I) = \int d\psi \langle \psi | I^\dagger \epsilon(|\psi\rangle\langle\psi|) I | \psi \rangle$, which runs over the entire Bloch sphere. Using this measure of the teleportation fidelity, we find that only one cross pair (qubits 0 and 5) and one straight pair (qubits 0 and 1) have fidelity exceeding $2/3$, the average of which can be achieved with classical communication alone without quantum correlations [53].

We ask whether it is possible to guarantee better than classical results for all pairs if we have additional information about the identity of the states. This problem has been considered by Roy *et al.* [27], who showed that the increased information about the states to be teleported obtained by restricting them to a polar cap of the Bloch sphere [see Fig. 10(a)] leads to an increase in the bound on the maximum fidelity obtainable by purely classical means. Here we look to exploit this and see if the gain in the average teleportation fidelity due to considering a small portion of the Bloch sphere having states that are transferred with better quality in our MQNC implementation is enough to raise it above the classical bound for all four pairs of the network code. The portion of the Bloch sphere that we consider is a polar cap, as in Ref. [27], but where the pole is rotated to coincide with the teleported state that has maximum fidelity for each pair and the fidelity of states near it are generally higher than states from the remainder of the Bloch sphere.

Instead of considering the reconstructed Choi matrix Λ for each channel using a pair, we use the Choi matrix to plot the teleportation fidelity for individual states on the surface of the Bloch sphere in Fig. 10(b) for the four different teleportation routes. Here, the fidelity is given by $F(\theta, \phi) =$

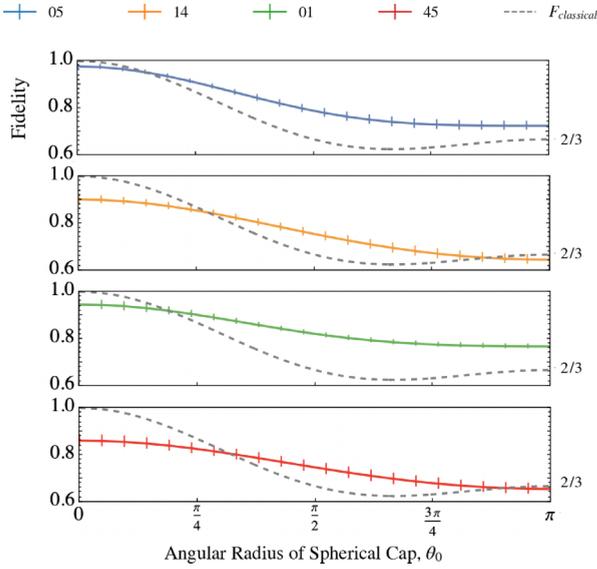


FIG. 11. The average teleportation fidelity of states from a spherical cap of the Bloch sphere. Error bars correspond to one standard deviation of the fidelity obtained from 30 reconstructed Choi matrices for each channel. The case where $\theta_0 = \pi$ corresponds to the average over the entire sphere. Dashed lines show the highest fidelity achievable through classical communication alone.

$\langle \psi | \epsilon(\rho_{in}) | \psi \rangle$, with $\rho_{in} = |\psi\rangle\langle\psi|$, $\epsilon(\rho_{in}) = \text{Tr}_1[\Lambda(\rho_{in,1}^T \otimes I_2)]$ and $|\psi\rangle = \cos(\theta/2)|0\rangle + e^{i\phi}\sin(\theta/2)|1\rangle$. From the fidelity plots, one can see that in all cases, there are well-defined areas of states which are teleported with relatively high fidelity, and that for each pair, these areas are stable over the entire 30 runs of the demonstration. Most of the high-fidelity states fall within a spherical cap (i.e., a rotated polar cap) of the Bloch sphere.

For each pair of the network code, we determine the point of highest teleportation fidelity, F_{\max} , and calculate the average fidelity by performing numerical integration over pure states falling within a certain distance of this point. The distance is determined so as to give states within a spherical cap of angular radius θ_0 . This is repeated for a range of angles. For each of the 30 channels obtained (for each pair), the average fidelity at a given angular radius is calculated by numerical integration as $F_{\text{ave}}(\epsilon, I, \theta_0) = \frac{1}{S} \int_0^{\theta_0} \int_0^{2\pi} F(\theta, \phi) \sin\theta d\theta d\phi$, where $S = \int_0^{\theta_0} \int_0^{2\pi} \sin\theta d\theta d\phi$. To take into account the rotated polar cap, the input state $|\psi\rangle$ is rotated by θ_{\max} and ϕ_{\max} , before and after the channel, to align the state $|0\rangle$ along the axis of maximum fidelity, F_{\max} , as shown in Fig. 10(b) for the four different teleportation routes. One can see from Fig. 11 that for all pairs, the classical bound [27] $F_{cl}(\theta_0) = 1 - [(2 + \cos\theta_0)(1 - \cos\theta_0)]/6$ is exceeded for a large range of θ_0 . This demonstrates that quantum correlations were used in the transfer of the quantum states over all four channels using MQNC.

It is interesting to note that the average teleportation fidelity in Fig. 11 is different from the fidelity of the actual entangled pairs that are used, as shown in Fig. 9. In the case of the latter, the fidelity is a fixed value quantifying the link quality, while the former can change depending on the size and shape of the

distribution taken from the Bloch sphere. A higher value of the average teleportation fidelity than the entangled pair fidelity does not represent a contradiction because they represent two different aspects in terms of the channel quality. As the distribution is reduced around the point of maximum fidelity of the Bloch sphere (from Fig. 10), it is expected that the average fidelity of the smaller distribution will increase, and its value is less impacted by the fidelity of the entangled link.

An interesting example is the extreme case where the unentangled state $|0\rangle|+\rangle$ is used as the pair, which has a fidelity of 0.5 with respect to the two-qubit graph state. However, when using it to teleport states near to the state $|0\rangle$, i.e., a small polar cap, the average teleportation fidelity approaches unity. Further details on the average teleportation fidelity for the polar cap distribution, and its relation to the entangled pair fidelity and concurrence, can be found in the theory paper by Roy *et al.* [27].

D. Noise

Our limited success in the preceding section raises questions regarding the sources of error and what may be needed to overcome these so that arbitrary states may be teleported. Unfortunately, the current state of the IBM Quantum processors is such that errors are highly variable to the point where even hourly recalibration does not result in consistent behavior over a span of minutes. Since we are primarily concerned here with a repeatable demonstration that MQNC is already somewhat feasible on these processors, we have compensated for the variable errors by taking averages of the results spanning multiple recalibrations. It makes sense then to consider an averaged error model when discussing such demonstrations.

In Ref. [54], we explored the efficacy of a depolarizing noise model for our MQNC implementation, and were able to predict the density matrices of the MQNC qubit pairs with $\sim 90\%$ fidelity. This suggests that a depolarizing channel model of noise represents well the collective impact of the error sources on the protocol in the IBM processor, complementing the results of Ref. [26] and serving as a starting point for ballpark estimates or comparison to similar demonstrations on IBM processors.

On the other hand, predictions regarding the performance of future quantum processors are better made with reference to improvements of the physical hardware and consequent reduction in specific kinds of error. We can reasonably expect that single- and two-qubit gate errors, readout errors, and T1 and T2 decoherence all contribute to the total error which is propagated to the final teleported state, so it would be sensible to study the expected effects of error in a controlled way using QISKIT's noisy quantum assembly language (QASM) simulator. Given the small number of involved qubits, this is a reasonable proposition in terms of the required computational resources, but since the analysis of such a study is nontrivial, this is left for future work.

V. MAPPING THE MQNC SWITCH ONTO IBM Q PROCESSORS

We have demonstrated teleportation over MQNC on a single butterfly network, giving an indication of the degree of

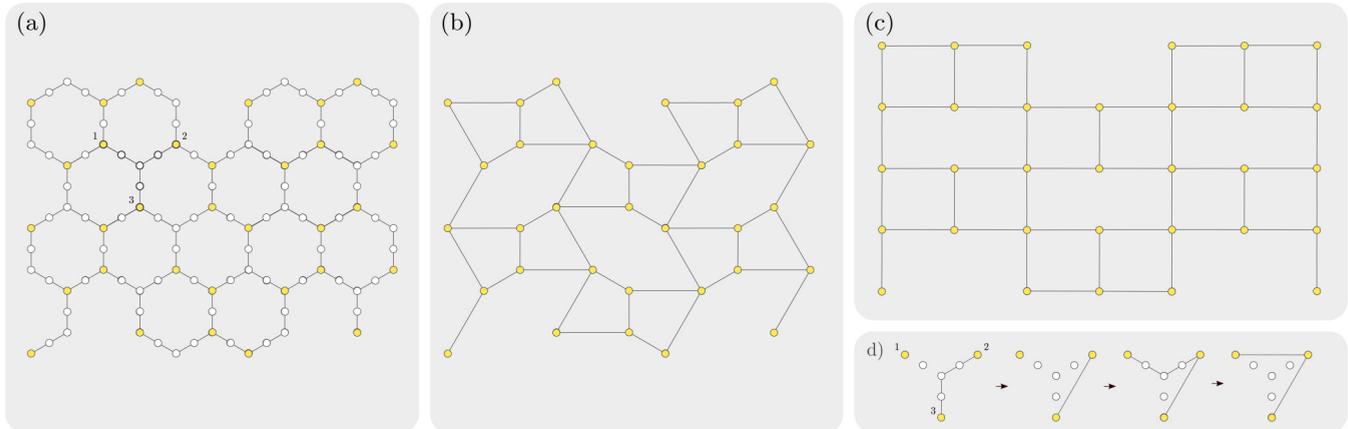


FIG. 12. Embedding of the 4×4 MQNC switch in the IBM processors. (a) The heavy-hex layout. Switch qubits are represented as yellow dots. White dots represent auxiliary qubits used in the creation of the switch. Edges represent nearest-neighbor relations. (b) The MQNC switch graph state. Edges represent entanglement. (c) The same graph state as in (b), rearranged to show the gridlike layout. (d) The procedure for forming additional connections to a qubit.

feasibility of MQNC on superconducting quantum processors at present. Considering the high degree of noise present even in this simple case, implementation of the MQNC switch is unfortunately not yet practical. Eventually, error correction will mitigate the effects of noise either completely or to a large extent, but in the interim we may still see significant reduction in noise, which would enable implementation of the MQNC switch with some degree of success even in the absence of error correction. This would be of interest from the standpoint of demonstrating the feasibility of the MQNC switch and potential NISQ applications using it, and therefore we discuss how the switch may be embedded in a processor utilizing the heavy-hex layout.

In Fig. 12, we present an embedding of the MQNC switch in the heavy-hex layout [shown explicitly in Fig. 12(a)] that makes use of a subset of the available qubits, which forms a tessellating pattern [shown in Fig. 12(b)]. This pattern is identical to the one in Fig. 12(c) and the generalized MQNC previously shown in Fig. 5(a). The qubits that are used are not adjacent, and so it is necessary to use graph-state rewiring to entangle them. Given two qubits which are connected via an isolated path in the heavy-hex layout, the two qubits may be directly connected by entangling the qubits of the path via CZ's and subsequently measuring all the qubits in the path apart from the ends in the Y -basis—following transformation rule T2. The majority of the connections shown in Fig. 12(b) are created in this way by performing Y -measurements on the white qubits in Fig. 12(a). However, it is not possible to create all the required connections using a single round of Y -measurements since not all white qubits lie on isolated paths. The qubits labeled 1, 2, and 3 in Fig. 12(a) are an example of this. In order to connect qubit 2 to both qubit 1 and qubit 3, the connections must be formed in two separate Y -measurement steps, as shown in Fig. 12(d). Using this procedure, it becomes possible to have up to four connections per qubit despite the heavy-hex layout permitting, at most, three direct connections per qubit.

Note that the qubits which are measured out could potentially be used again to verify the presence of the desired connections using a parity checking method similar to that in

[55], with only minor adjustments being required to translate their Greenberger-Horne-Zeilinger (GHZ)-state technique to our graph states—at the cost of added complexity. This appears promising, but will likely require some optimization to be used effectively, and so we leave an investigation of it for future work where it may be treated with appropriate subtlety.

VI. DISCUSSION

Small-scale quantum communication networks will likely become feasible within the next few years as the technology continues to improve; however, the NISQ regime presents challenges in the form of limited channel capacity and a limited number of qubits so that efficient use of these resources and their characterization is paramount [10, 11].

The question of how practical MQNC might turn out to be in a realistic quantum communication network depends on many different aspects. For instance, when the physical resources are limited in a network, and multiple qubits are available for performing purification, these qubits could be used directly for solving contention. Furthermore, without a high-quality quantum memory available at network nodes and with probabilistic remote entanglement generation, the cost of pregeneration of the resource state could be very large. In addition, the topology of the network could limit the shape of the MQNC resource state in the multiparty case and one would need to adapt the MQNC multiparty state that we have proposed in Sec. II C to that of the network topology, and determine whether the adapted state can be transformed into independent Einstein-Podolsky-Rosen (EPR) pairs for teleportation between desired nodes, which is an NP-complete problem [56]. Thus, MQNC may provide an efficient and scalable solution to contention-free switching in quantum networks, potentially increasing network throughput compared to routing, but this depends on a number of practical aspects, including the allowed network topology.

In this paper, we presented a successful demonstration of MQNC combined with quantum teleportation exceeding the maximum fidelity achievable by classical means. Our work builds on a prior demonstration wherein MQNC was studied

on an older IBM superconducting processor, while introducing a number of innovations. By introducing a rewiring scheme based on local complementation, we are able to create the MQNC resource state on newer IBM processors, whereas standard SWAP-based transpiling on this topology introduces too much noise for creation of the state. Using the new processors, we were able to establish sufficient quantum correlations for teleportation of quantum information. We found, however, that the average teleportation fidelity for states sampled from the entire Bloch sphere only exceeded the classical bound for teleportation across one of the two final routes generated using MQNC (in either the crosswise or straight configuration). Some states were teleported with disproportionately low fidelity and the average teleportation fidelity considerably increased when taken over a distribution excluding these states.

By adapting recent theory work on teleportation from nonuniform distributions, we were able to show that the classical bound was exceeded in the case of a spherical cap distribution even when the increased classical knowledge about the state is accounted for. This likely extends to any teleportation scheme with biased noise. We also introduced a generalization of MQNC to an arbitrary number of source-destination pairs which requires few connections per qubit and is therefore suitable for 2D quantum processor architectures.

Our results show that high-fidelity teleportation of quantum states using MQNC on a real superconducting processor is possible, but only if the states belong to a subset of the Bloch sphere. Nonetheless, we have shown that it is already possible to transfer quantum information over the MQNC, which opens up the possibility of using entanglement purification to obtain a high-fidelity communication channel if several noisy ones are available [57,58]. Furthermore, there exist protocols such as the BB84 QKD protocol [59], which rely on repeat-until-success strategies and can therefore make use of an imperfect communication channel. Lastly, we note that while current processors are still too noisy for truly practical MQNC, the incorporation of error correction is planned for future IBM processors [29]. This has the potential to negate

the majority of the effects of noise in the protocol and there is the added advantage of the logical topology supporting direct creation of the MQNC switch via encoded graph states. The fact that present-day superconducting quantum processors are already capable of implementing butterfly MQNC with a relatively high degree of success suggests that MQNC presents a practical solution to contention in future NISQ quantum communication networks.

ACKNOWLEDGMENTS

We acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors, and do not reflect the official policy or position of IBM or the IBM Quantum team. We thank Taariq Surtee and Barry Dwolatzky at the University of Witwatersrand and Ismail Akhalwaya at IBM Research Africa for access to the IBM processors through the Q Network and Africa Research Universities Alliance. This research was supported by the Department of Science and Innovation (DSI) through the South African Quantum Technology Initiative (SA QuTI), Stellenbosch University, the National Research Foundation (NRF), and the Council for Scientific and Industrial Research (CSIR).

APPENDIX: CALIBRATION PARAMETERS

The demonstration was carried out on `ibm_cairo` version 1.0.24 between 9:00 p.m. on April 26, 2022 and 5:00 a.m. on April 27, 2022 (UTC). Table I lists calibration parameters for this processor during the daily calibration performed by IBM Quantum on April 27, 2022 at approximately 8:00–9:00 a.m. (UTC). The layout of the processor is given in Fig. 6. For the sake of rough comparison, we include Table II, which lists IBM Q20 Tokyo properties at around the time of the experiment of Pathumsoot *et al.*, but this is of limited relevance since the need for rewiring means that the improvements in `ibm_cairo` do not directly translate to improvements in MQNC performance.

TABLE I. Calibration parameters from daily calibration of `ibm_cairo` as reported by IBM Quantum. Last update date is April 27, 2022 at 9:10 a.m. (UTC).

Qubit	2	3	5	8	9	11	14	13
T1 (μ s)	9.03×10^1	1.64×10^2	1.02×10^2	9.74×10^1	8.30×10^1	7.43×10^1	8.16×10^1	1.11×10^2
T2 (μ s)	7.26×10^1	1.95×10^2	6.63×10^1	7.85×10^1	5.21×10^1	1.25×10^2	1.11×10^2	1.78×10^2
Frequency (Hz)	4.91×10^9	5.12×10^9	5.05×10^9	4.97×10^9	5.23×10^9	5.13×10^9	5.04×10^9	5.28×10^9
Anharmonicity (Hz)	-3.44×10^8	-3.40×10^8	-3.41×10^8	-3.43×10^8	-3.40×10^8	-3.42×10^8	-3.42×10^8	-3.39×10^8
Readout error	1.16×10^{-2}	7.90×10^{-3}	7.40×10^{-3}	1.25×10^{-2}	3.58×10^{-2}	7.30×10^{-3}	6.90×10^{-3}	6.80×10^{-3}
Prob meas. 0 prep. 1	1.40×10^{-2}	1.00×10^{-2}	8.80×10^{-3}	1.74×10^{-2}	3.18×10^{-2}	9.20×10^{-3}	7.80×10^{-3}	7.60×10^{-3}
Prob meas. 1 prep. 0	9.20×10^{-3}	5.80×10^{-3}	6.00×10^{-3}	7.60×10^{-3}	3.98×10^{-2}	5.40×10^{-3}	6.00×10^{-3}	6.00×10^{-3}
Readout length (s)	7.32×10^{-7}							
ID error	1.606×10^{-4}	1.937×10^{-4}	2.354×10^{-4}	2.938×10^{-4}	2.525×10^{-4}	2.038×10^{-4}	1.644×10^{-4}	2.010×10^{-4}
\sqrt{X} error	1.606×10^{-4}	1.937×10^{-4}	2.354×10^{-4}	2.938×10^{-4}	2.525×10^{-4}	2.038×10^{-4}	1.644×10^{-4}	2.010×10^{-4}
Pauli-X error	1.606×10^{-4}	1.937×10^{-4}	2.354×10^{-4}	2.938×10^{-4}	2.525×10^{-4}	2.038×10^{-4}	1.644×10^{-4}	2.010×10^{-4}
Qubit pair	(2 3)	(3 5)	(5 8)	(8 11)	(8 9)	(11 14)	(14 13)	
CX error	8.89×10^{-3}	4.51×10^{-3}	5.55×10^{-3}	8.54×10^{-3}	1.37×10^{-2}	9.24×10^{-3}	4.48×10^{-3}	

TABLE II. Calibration parameters from IBM Q20 Tokyo on 29 August 2019 at 7:00 PM (UTC) as reported by IBM Quantum.

Qubit	0	1	5	6	10	11	
T1 (μ s)	1.230×10^2	6.954×10^1	7.037×10^1	7.713×10^1	1.123×10^2	7.378×10^1	
T2 (μ s)	8.261×10^1	1.066×10^1	5.538×10^1	6.279×10^1	6.371×10^1	6.221×10^1	
Frequency (GHz)	5.020×10^0	4.903×10^0	5.143×10^0	5.033×10^0	4.960×10^0	5.230×10^0	
Readout error	6.000×10^{-2}	4.400×10^{-2}	4.300×10^{-2}	3.100×10^{-2}	1.020×10^{-1}	3.600×10^{-2}	
Prob. meas. 0 prep. 1	6.200×10^{-2}	8.000×10^{-2}	5.200×10^{-2}	5.200×10^{-2}	1.780×10^{-1}	4.000×10^{-2}	
Prob. meas. 1 prep. 0	5.800×10^{-2}	8.000×10^{-3}	3.400×10^{-2}	1.000×10^{-2}	2.600×10^{-2}	3.200×10^{-2}	
Qubit pair	(0 1)	(0 5)	(1 6)	(5 6)	(5 10)	(6 11)	(10 11)
CX error	3.52×10^{-2}	4.99×10^{-2}	3.43×10^{-2}	2.29×10^{-2}	2.64×10^{-2}	1.65×10^{-2}	3.39×10^{-2}

- [1] R. Van Meter and S. J. Devitt, *Computer* **49**, 31 (2016).
- [2] N. Gisin and R. Thew, *Nat. Photon.* **1**, 165 (2007).
- [3] S. Wehner, D. Elkouss, and R. Hanson, *Science* **362**, eaam9288 (2018).
- [4] X.-M. Jin *et al.*, *Nat. Photon.* **4**, 376 (2010).
- [5] Y.-A. Chen *et al.*, *Nature (London)* **589**, 214 (2021).
- [6] M. Pompili *et al.*, *Science* **372**, 259 (2021).
- [7] P. Magnard *et al.*, *Phys. Rev. Lett.* **125**, 260502 (2020).
- [8] M. Mirhosseini, A. Sipahigil, M. Kalaei, and O. Painter, *Nature (London)* **588**, 599 (2020).
- [9] P. Das, S. K. Vittal, and M. Qureshi, [arXiv:2204.13142](https://arxiv.org/abs/2204.13142).
- [10] E. Tham, I. Khait, and A. Brodutch, [arXiv:2206.09938](https://arxiv.org/abs/2206.09938).
- [11] F. L. Buessen, D. Segal, and I. Khait, *Phys. Rev. Res.* **5**, L022003 (2023).
- [12] A. Eddins, M. Motta, T. P. Gujarati, S. Bravyi, A. Mezzacapo, C. Hadfield, and S. Sheldon, *PRX Quantum* **3**, 010309 (2022).
- [13] S. Perseguers *et al.*, *Rep. Prog. Phys.* **76**, 096001 (2013).
- [14] G. Chiribella, G. M. D'Ariano, and P. Perinotti, *Phys. Rev. A* **80**, 022339 (2009).
- [15] H. J. Kimble, *Nature (London)* **453**, 1023 (2008).
- [16] J. Preskill, *Quantum* **2**, 79 (2018).
- [17] M. Hayashi, *Phys. Rev. A* **76**, 040301(R) (2007).
- [18] H. Kobayashi, F. Le Gall, H. Nishimura, and M. Rötteler, in *Automata, Languages and Programming*, edited by S. Albers *et al.* (Springer, Berlin, 2009), pp. 622–633.
- [19] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, *IEEE Trans. Inf. Theory* **46**, 1204 (2000).
- [20] T. Ho and D. Lun, *Network Coding: An Introduction* (Cambridge University Press, Cambridge, 2008).
- [21] S.-Y. Li, R. W. Yeung, and N. Cai, *IEEE Trans. Inf. Theory* **49**, 371 (2003).
- [22] M. Hayashi *et al.*, in *STACS 2007*, edited by W. Thomas and P. Weil (Springer, Berlin, 2007), pp. 610–621.
- [23] T. Satoh, F. Le Gall, and H. Imai, *Phys. Rev. A* **86**, 032331 (2012).
- [24] H. Lu *et al.*, *npj Quantum Inf.* **5**, 89 (2019).
- [25] T. Matsuo, T. Satoh, S. Nagayama, and R. Van Meter, *Phys. Rev. A* **97**, 062328 (2018).
- [26] P. Pathumsoot, T. Matsuo, T. Satoh, M. Hajdusek, S. Suwanna, and R. Van Meter, *Phys. Rev. A* **101**, 052301 (2020).
- [27] S. Roy, S. Mal, and A. Sen(De), *Phys. Rev. A* **105**, 022610 (2022).
- [28] F. Arute *et al.*, *Nature (London)* **574**, 505 (2019).
- [29] C. Chamberland, G. Zhu, T. J. Yoder, J. B. Hertzberg, and A. W. Cross, *Phys. Rev. X* **10**, 011022 (2020).
- [30] B. A. Huberman and B. Lund, *Inf. Syst. Front.* **22**, 37 (2020).
- [31] M. Hein, J. Eisert, and H. J. Briegel, *Phys. Rev. A* **69**, 062311 (2004).
- [32] R. Raussendorf and H. J. Briegel, *Phys. Rev. Lett.* **86**, 5188 (2001).
- [33] H. J. Briegel *et al.*, *Nat. Phys.* **5**, 19 (2009).
- [34] D. Leung, J. Oppenheim, and A. Winter, *IEEE Trans. Inf. Theory* **56**, 3478 (2010).
- [35] X.-B. Pan, G. Xu, Z.-P. Li, X.-B. Chen, and Y.-X. Yang, *Quantum Inf. Process.* **20**, 65 (2021).
- [36] W. K. Wootters and W. H. Zurek, *Nature (London)* **299**, 802 (1982).
- [37] R. Ratan, M. K. Shukla, and A. Y. Oruç, in *2007 41st Annual Conference on Information Sciences and Systems*, edited by G. G. L. Meyer and H. Weinert (IEEE, New Jersey, 2007), pp. 789–793.
- [38] R. A. Spanke and V. Benes, *Appl. Opt.* **26**, 1226 (1987).
- [39] C. Chang and R. Melhem, *Parallel Process. Lett.* **07**, 279 (1997).
- [40] V. E. Beneš, *Bell Syst. Tech. J.* **43**, 1619 (1964).
- [41] D. Schlingemann and R. F. Werner, *Phys. Rev. A* **65**, 012308 (2001).
- [42] D. Schlingemann, *Quantum Inf. Comput.* **2**, 307 (2002).
- [43] M. Grassl, A. Klappenecker, and M. Rötteler, in *Proceedings 2002 IEEE International Symposium on Information Theory (ISIT 2002), Lausanne, Switzerland* (IEEE, 2002), p. 45.
- [44] A. M. Childs, E. Schoute, and C. M. Unsal, in *14th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2019)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 135 (Leibniz-Zentrum für Informatik, Schloss Dagstuhl, 2019), pp. 3:1–3:24.
- [45] G. Li, Y. Ding, and Y. Xie, in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, edited by I. Bahar, M. Herlihy, E. Witchel, and A. R. Lebeck (Association for Computing Machinery, 2019), pp. 1001–1014.
- [46] G. J. Mooney, C. D. Hill, and L. C. Hollenberg, *Sci. Rep.* **9**, 13465 (2019).
- [47] G. Tóth and O. Gühne, *Phys. Rev. Lett.* **94**, 060501 (2005).
- [48] G. Tóth, *Comput. Phys. Commun.* **179**, 430 (2008).

- [49] D. F. James *et al.*, in *Asymptotic Theory of Quantum Statistical Inference: Selected Papers*, edited by M. Hayashi (World Scientific, Singapore, 2005), pp. 509–538.
- [50] E. Størmer, *Positive Linear Maps of Operator Algebras* (Springer, New York, 2013), pp. 49–62.
- [51] S. Massar and S. Popescu, *Phys. Rev. Lett.* **74**, 1259 (1995).
- [52] S. Massar and S. Popescu, in *Asymptotic Theory Of Quantum Statistical Inference: Selected Papers* (World Scientific, Singapore, 2005), pp. 356–364.
- [53] R. Horodecki, M. Horodecki, and P. Horodecki, *Phys. Lett. A* **222**, 21 (1996).
- [54] H. Rall and M. S. Tame, in *The Proceedings of SAIP2022, the 66th Annual Conference of the South African Institute of Physics*, edited by A. Prinsloo (South African Institute of Physics, 2022).
- [55] G. J. Mooney *et al.*, *J. Phys. Commun.* **5**, 095004 (2021).
- [56] A. Dahlberg, J. Helsen and S. Wehner, *Quantum* **4**, 348 (2020).
- [57] C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, and W. K. Wootters, *Phys. Rev. Lett.* **76**, 722 (1996).
- [58] D. Deutsch, A. Ekert, R. Jozsa, C. Macchiavello, S. Popescu, and A. Sanpera, *Phys. Rev. Lett.* **77**, 2818 (1996).
- [59] C. H. Bennett and G. Brassard, *Theor. Comput. Sci.* **560**, 7 (2014).